Performance Considerations for High-End PC A/D Boards

by Larry Copeland, Datel Inc.

Today's digital signal processing (DSP) and Fast Fourier Transform (FFT) applications using analog-to-digital, digital-to-analog (A/D, D/A) boards installed in PCs are pushing the system limits. The newer high-end PC's are not really designed as instrument platforms, but their high performance and low prices force many researchers to seriously consider them - even if they would prefer a real data acquisition system.

Most A/D and D/A board applications, especially in the manufacturing, industrial and process control industries, have sample rates much slower than the host PC. Users just plug in the boards and things take care of themselves.

Newer applications are demanding much higher sample rates as digital recording displaces older all-analog methods. High-speed DSP techniques have opened a new world of test and manufacturing applications. We'll only discuss fast analog sample rates (500kHz or more) where the PC speed *is* a limitation. Current A/D boards offer blinding speed. Some can deliver nonstop 10MHz A/D rates to the Extended Industry Standard Architecture (EISA) bus if your system can take it.

The PC applies the brakes in several ways. The burning question for most high-speed A/D user is, "How fast can it go?". That depends on how the application is structured. But are you really sure you need to go that fast?

Time Architecture

The timing architecture of a high-speed A/D application may be analyzed by preparing a chart or milestone diagram showing all the major events in collecting and using the data. Timing breaks down into two broad categories:

- Real-time (concurrent) A/D sampling
- Non-real-time (post-processing) A/D sampling

In real-time applications, A/D data is processed without stopping the A/D sampling. In contrast, non-real-time has two distinct phases: (1) store the data and (2) stop the A/D, then use the data. Some minimal concurrent processing may be required just to get the data to the storage device.

The big distinction occurs when the processed data is used. The highest A/D sample rates almost always require post-processing because there's no time for processing during data acquisition. Some A/D applications do not periodically stop the A/D sampling with each trigger. These let the A/D continuously convert and are called nonstop "streaming" or "gapless" recording.

Often users look at their low-speed industrial application and think they don't have a fast A/D problem. But consider that a 64-channel industrial A/D system sampling at only 10kHz/channel is really producing data at 640kHz - too fast for most disk systems.

Local A/D Board Storage

The A/D usually runs at a constant clock rate or a repeating periodic triggered clock, whereas the PC runs in block bursts of data. The PC is not very efficient if it takes only single samples. Memory buffering on the A/D board decouples these two timing processes.

All fast (500kHz or higher) A/D boards have some local memory to store samples temporarily until the host PC can take them. These systems have two design requirements:

- Keep the A/D running nonstop while the host PC is briefly unavailable.
- Provide dual-port memory, which enables the simultaneous streaming feature.

No More Dumb Boards

All boards for high-speed applications must be able to run unattended without constant nagging by the host PC. A local microcomputer is sometimes used if heavy math preprocessing is needed, such as DSP. Usually, though, a discrete logic controller or state machine is actually faster than a CPU. Remember, the religion here is speed (Figure 1).

Reprinted with permission from EE - Evaluation Engineering, February 1995

Oversample Errors and Testing

Oversampling occurs when previous samples are not read from the board fast enough before the A/D generates new samples. This can be flagged to the user by a FIFO overflow signal or some logic which knows if a new A/D clock is started before the last A/D sample was read into the buffer.

If you want to be absolutely sure how fast your board runs, test it in your application. We recommend that you collect many samples (several megabytes), check for overflow errors, then increase the A/D sample clock and rerun the test. At some clock speed , you will get overflow errors, indicating lost data because the PC can't keep up. You must use large amounts of memory because too small a memory will not detect all the speed limits.

Buses, CPUs and Memory

A/D data which flows into system memory must pass through the backplane bus of the PC. On Industry Standard Architecture (ISA) bus computers (the original 16-bit PC/AT bus), this bus can transfer at most about 1.2 16-bit megawords/s, 2.4MB/s in a burst, even with a 90MHz Pentium® installed. EISA systems can transfer data at up to 32MB/s in a 1k mode C DMA burst.

System memory uses a fast private bus to the CPU which is not on the backplane system bus. But there are two phases to moving data from the A/D board: reading from the board, then writing to memory. The transfers never go direct from the A/D board to system memory. The *reading* portion must use the backplane bus and goes through bus cycle delays.

Adding a faster CPU will increase the system speed, but there are limits imposed for any transfers that go over the backplane. For example, going from a 32MHz 80486 to 90MHz Pentium will have practically no measurable effect because the bus is already going as fast as it can. The only place you can see the improvement is with activity which is purely in memory, because it stays off the bus.

They Didn't Build Windows® for Us

Windows offers two features for advanced data acquisition - extended memory management and multiple simultaneous processes (multitasking). Extended memory is organized (as descriptors) somewhat differently than conventional base memory below the 1MB address. You must access the extended memory through Basic Input/Output Systems (BIOS) calls that take extra time because the CPU must be in a protected mode and because of descriptor overhead.

Under Windows, several applications can be started and seemingly all run at the same time using one rapidly switched CPU. Windows programs must be specially designed to do this. Each program runs by itself for some time, then returns control to Windows, which then runs the next program on its list. Windows eventually returns to your program; but if some other program hogs Windows, you might wait quite a while.

You can fool Windows by running in the locked mode, not releasing control by not accessing the Windows message loop for awhile. The screen will remain fixed and the mouse will have no effect. This is not considered good Windows programming form.

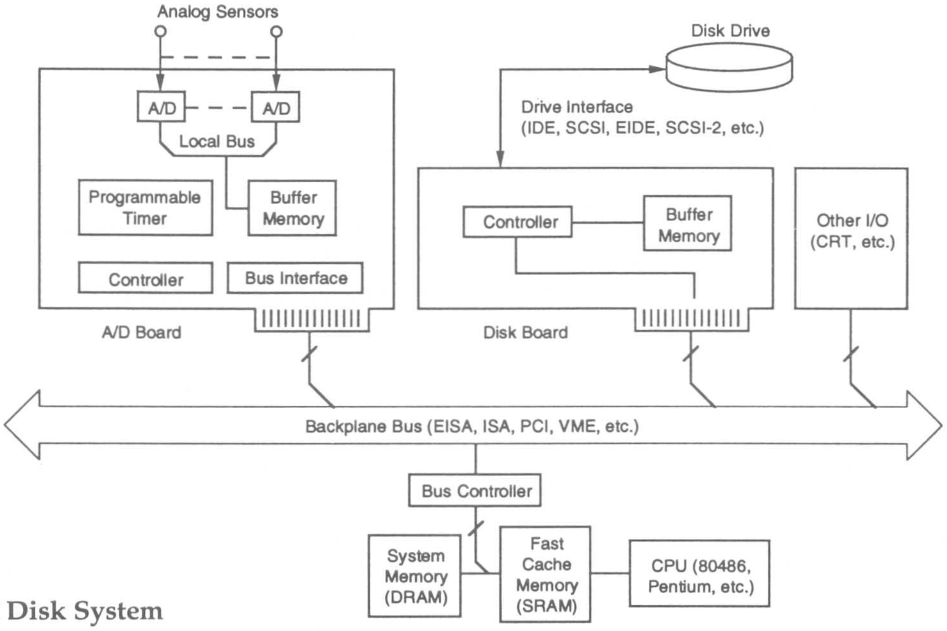

Figure 1. A/D PC Disk System

Windows is a Microsoft Corp. trademark
Pentium is an Intel Corp. trademark

Windows NT and Windows 95 are supposed to be preemptive, providing a known, predictable response time. But the response still might not be very fast.

This sequential multitasking means that you cannot say for sure when Windows will run your program again or how fast you can collect data. For rather slow A/D collection, there is no problem. But for fast sampling, Windows may run for awhile, then start missing data. You can improve the speed by shutting off all other unnecessary applications. But there is a limit.

Windows also uses the disk as virtual memory if it cannot find real physical memory. Since disk is much slower than memory, extraordinary delays can be generated. Windows doesn't tell you it's doing this. But that winking disk drive light is a dead giveaway. These tasks combine in random ways and all require system resources and extra time.

Real-time software programmers call these characteristics probabilistic or nondeterministic, meaning that you cannot predict the speed of Windows with absolute certainty. What you need is a preemptive operating system which responds in a fast, guaranteed minimum time. Low-speed A/D applications will never see this effect. It is unique to high-end A/D.

How can Windows be this way? Simple. Windows was not designed for real-time high-speed A/D conversion. It's only an accident that it works at all. Microsoft was not thinking about scientific and instrumentation users when they wrote the system.

Windows was built for banks, insurance companies and businesses. It's great for MIS and EDP departments where they don't notice that the system is unavailable for 1/10s. But that's 100,000μs - a virtual eternity to a fast A/D process. In that time, we may have lost 100,000 A/D samples from a 1MHz A/D converter.

Even if we somehow temporarily store that data on the A/D board, Windows has to go even faster to catch up. A/D board manufacturers know how to get the most out of Windows, but DOS software will always be somewhat faster. And don't try to run your fast DOS A/D software under the Windows MS-DOS prompt. It won't improve.

A/D Data Disk Storage

Almost all high-end PC's include disks, so it is natural to consider them as fast A/D data recorders. Like Windows, disks were not designed for A/D usage.

The disk system consists of the mechanical disk drive mechanism, the drive electronics (head positioning and spindle speed controller), the bus interface controller electronics and the BIOS/DOS software in the host memory. Clever software combined with a buffered autorunning A/D board can adapt the disk as an analog recorder. Multiple low-level BIOS software calls make all disk systems adaptable and interchangeable with the DOS.

For high-speed A/D, there are two methods for using the hard disk:

- **Disk only** - save data to disk as the A/D runs.

- **Memory then disk** - save data to large memory, then store it to disk after A/D sampling stops.

The disk-only method is much slower, but can use the total size of the disk drive (minus the DOS and other programs). The disk-only method can use either DOS-based disk operation or the faster direct-to-track method, not using the DOS. We will discuss only the DOS/BIOS method since it is designed for wide portability and easy recovery of the A/D data as binary files.

The memory-then-disk method uses the full speed available in the host PC, but is limited to the size of that memory. For truly large memory capacity, you may need a computer with EISA or Peripheral Component Interconnect (PCI) buses, although a few high-end ISA 80486s are now getting more than 16MB.

Some EISA machines can accept 256MB of DRAM or about 125 megasamples of A/D data. At 1MHz sample rates, that's only two minutes of data recording. But for someone instrumenting an automobile crash test, that may be plenty.

Two immutable delays show up in the hard disk mechanical system. One is the seek delay where the read/write head physically moves to another cylinder after using the previous cylinder. Seeks average 3ms to 10ms on fast disks between adjacent cylinders. During the seek, you cannot access the disk.

Rotational latency delays involve the mechanical time for a selected sector to fall under the read/write head so that data recording can begin. At worst case, you may have just missed the sector and must wait almost a full disk revolution to start writing. Most hard disks rotate at 3,600 to 6,000 rpm or almost 16.7ms to start writing. Once again, disk access is stopped while the latency delay finishes.

Once the sector is accessed, the drive controller rapidly transfers large amounts of data to sequential disk sectors on the same cylinder. Then comes another dreaded seek delay and probably a latency delay. One cylinder might hold 50kB to 200kB (depends on how many sides there are). So the seeks will not come too often. The seek delay is guaranteed. Latency delays are random but have a maximum delay of one revolution, assuming no media errors.

New bus interfaces such as the fast, wide Small Computer System Interface (SCSI) SCSI-3 or enhanced integrated disk electronics (EIDE) mean that the block transfer time drops out of the equation. The seeks and latency delays remain.

Disk drive controller manufacturers strive to hide or postpone these delays. Usually, this means designing in large memory buffers (some up to 0.5MB) on the disk controller.

The system will attempt its own sector writes and seeks if there is still data left in the controller's buffer. You can request a disk write. The A/D data blasts over to the disk controller, the BIOS disk call returns quickly and it appears you are ready to send more data. Meanwhile, the controller tries to write out your data onto the disk.

This goes on for many more blocks, then the BIOS call does not immediately return. Your CPU is spinning, waiting for the disk controller to say done. The disk buffer is full and the disk controller is waiting for a seek or latency delay to finish. Now your A/D board has nowhere to send its high-speed data (Figure 2.)

There is no way to avoid these two delays on a single disk drive. But you can program around them with several disks in swapped ping-pong or round-robin fashion, then carefully predict when seeks will happen. Even redundant array of independent disks (RAID) systems may not avoid these seek captures so they, too, might need special code. So how fast are current disks? Single ISA IDE drives seem to be from 75kHz to 100kHz. Wide fast SCSI-2/3 on EISA or PCI drives are from 250kHz to 400kHz. RAID systems are probably faster.

Myths and Answers

Now that we've discussed the PC's major components in the speed equation, let's look at some common misconceptions:

Direct Memory Access (DMA) is the fastest way to transfer data.

A qualified maybe. If you have a 80486 clone, program transfers are probably faster than DMA. Now, under the EISA bus, mode C DMA transfers move a 32-bit doubleword every 125ns in a 1k burst. DMA on the PC only moves data between memory and I/O - not memory-to-memory. If your A/D board is memory mapped, EISA DMA is out.

The DMA controller requires real physical addresses. It cannot handle protected-mode descriptor addresses until they are converted to absolute addresses. This adds delays using extended memory. Virtual DMA under Windows is also slower.

A Pentium makes my bus go fast; or, I should up upgrade from a 32MHz CPU to a 66MHz CPU; or, I should get a DX 80486 - that will give me high speed.

This is true for movements within memory only. Any transfers which use the backplane ISA bus show no improvement with fast 80486s and Pentiums.

I have an oscilloscope A/D board that goes 40MHz. Can I use it to digitize RF signals?

Most scope boards have a small amount of very high-speed memory. Usually this memory is not simultaneous dual access so it cannot do streaming applications. Your scope board can look at RF signals only in very short bursts no larger than the A/D buffer.

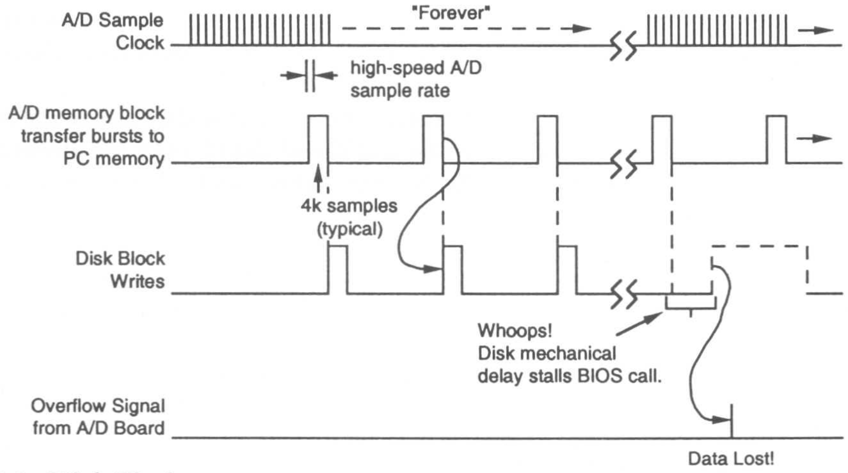
A/D Sample Clock
"Forever"
high-speed A/D sample rate
A/D memory block transfer bursts to PC memory
4k samples (typical)
Disk Block Writes
Whoops! Disk mechanical delay stalls BIOS call.
Overflow Signal from A/D Board
Data Lost!

Figure 2. A/D-to-Disk Timing

(Drawing is not to scale)

I'm running a 90MHz Pentium. I should have enough speed to save A/D data to disk and simultaneously do graphics screen writes.

Don't count on it. The Pentium mainly speeds up memory. It does nothing for slow video RAM, the much slower disk drive, and it will not speed up the slow ISA bus. Your best advice is to avoid screen or disk during high-speed A/D storage to memory.

Windows is faster than MS-DOS.

Absolutely not. Windows is slower than DOS for fast A/D; however, you can write code that runs Windows in a locked mode that is fairly fast.

I can make Windows faster with a graphics accelerator board.

This is true for A/D graphics placed on the screen. Accelerators improve some (not all) graphics.

The SCSI disk drive interface is faster than IDE.

Old 8-bit SCSI is fairly slow, especially if you have multiple devices, such as printers or LAN's, hanging off your 8-bit SCSI bus. Wide/fast SCSI-2/3 is faster than IDE, but enhanced IDE (EIDE) will give it a run for the money.

The new disk controllers with big caches take care of any disk-delay problems.

They help, but don't eliminate it. Sample long and fast enough and you'll find the mechanical speed limit of your drive.

SMARTDRV will take care of any disk-delay problems.

No, SMARTDRV will make it worse. SMARTDRV was designed for multiple read/write operations found in spreadsheets and data bases. It actually adds overhead for A/D disk recording.

My ISA A/D board is 2Mhz. You tell me my 90MHz Pentium can't take the data?

It can if it's EISA or PCI bus. Your ISA machine is good for, perhaps, 1MHz. You can, however, fill your A/D board's local memory at 2MHz. Then you must stop.

EISA automatically runs at 32MB/s.

Not if your DRAM memory controller needs extra CPU wait states. Some EISA systems add several waits for slow, cheap DRAM. The only sure way is thorough system testing. You may have to dive in there with an oscilloscope on the bus lines or do software timing.

I have 70ns DRAM chips. I should be able to go about 14MHz, right?

Wrong. There are more propagation delays in the DRAM memory controller, address decoding and memory transceiver logic you didn't account for.

Checklist to Speed Your Fast A/D System

Hardware

Get a fast A/D board which has oversample error detection, with simultaneous sampling and upload. It also needs a local controller so it can continue running while the host PC is busy. Make sure the board is designed for a nonstop streaming mode.

For signal processing and DSP/FFT usage, make sure your A/D board has high signal quality, such as wide input bandwidth, low noise, excellent harmonic distortion and great linearity. Some monolithic A/Ds are poor here. They sample at 1MHz but have only 50kHz full-scale bandwidth.

Get a fast bus for your A/D board, such as EISA or PCI.

Shut off system caches. Remove all unnecessary boards. Delete all unused Terminate and Stay Resident programs and device drivers from your CONFIG.SYS and AUTOEXEC.BAT files.

Sample only as fast as you really need. Store only as many samples as the application requires.

Memory

Shut off the EMM386 device driver. It is only needed for expanded memory emulation or allocation of certain memory blocks.

Disks and software

Use a wide, fast SCSI-2 or SCSI-3 system or an EIDE system with EISA or PCI disk controllers.

Defragment the drive and optimize the sector interleave so that additional data blocks are physically adjacent. The Norton Utilities and other programs do this.

Store A/D data in the root directory, not a subdirectory. Consider precompressing A/D samples on a fast A/D DSP board to reduce the amount of disk data.

Shut off the SMARTDRV disk cache. It is only useful for multiple read/write applications, such as spreadsheets and data bases.

Seek expert help. For the highest speed, find a disk software guru who knows your disk controller and can write A/D data in the direct-to-track method. For high speed, consider a RAID disk system.

For the highest possible A/D speed, don't try to do anything else while collecting data to memory. You can even stop the real-time clock to prevent responding to its interrupts. Don't even move the mouse. But do continue DRAM memory refresh.

Shut off any system-saver software which tries to periodically save your memory image. Kill any clock-based screen savers. All these things steal time.

Use DOS, not Windows. Don't use the MS-DOS prompt in Windows.

Conclusion

Getting fast, no-loss A/D performance is no small task. You need special tricks if you are truly pushing the performance envelope. But the rewards for learning your system are superior performance for a fraction of the cost and size of only a few years ago.

About the Author

Larry Copeland is the Product Marketing Manager for Datel's Subsystem Group. His previous experience includes nuclear medicine software programming and test equipment design. Datel Inc. 11 Cabot Blvd., Mansfield, MA 02048-1194, (508) 339-3000.

DATEL®
INNOVATION and EXCELLENCE

DATEL, Inc. 11 Cabot Boulevard, Mansfield, MA 02048-1194
Tel: (508) 339-3000 / Fax: (508) 339-6356
For immediate assistance 1-800-233-2765

DATEL (UK) LTD. Tadley, England Tel: (01256)-880444
DATEL S.A.R.L. Montigny Le Bretonneux, France Tel: 1-34-60-01-01
DATEL GmbH Munchen, Germany Tel: 89-544334-0
DATEL KK Tokyo, Japan Tel: 3-3779-1031, Osaka Tel: 6-354-2025

DATEL makes no representation that the use of its products in the circuits described herein, or the use of other technical information contained herein, will not infringe upon existing or future patent rights. The descriptions contained herein do not imply the granting of licenses to make, use, or sell equipment constructed in accordance therewith. Specifications are subject to change without notice. The DATEL logo is a registered DATEL, Inc. trademark.